CLAIMS

What is claimed is:

1. In a communication network, a method of TCP state migration comprising the steps of:

a) establishing a TCP/IP communication session between a client computer and a first bottom TCP (BTCP) module located below a first TCP module in a first operating system at a front-end node, said front end node part of a plurality of web server nodes that form a web server cluster containing information, said TCP/IP communication session established for the transfer of data contained within said information;

b) handing off said TCP/IP communication session from said first BTCP module to a selected back-end node over a persistent control channel; and

c) migrating a first TCP state of said first BTCP module to said selected back-end node, and sending a second TCP state of said selected back-end node to said first BTCP module over said control channel.

2. The method as described in Claim 1, wherein said step a) at said first BTCP module comprises the steps of:

a1) receiving a TCP/IP SYN packet from said client;

a2) selecting a first initial sequence number (ISN) for said first BTCP module that is associated with said TCP/IP communication session, said first ISN associated with a first TCP state of said first BTCP module;

a3) sending a TCP/IP SYN/ACK packet to said client;

a4) receiving a TCP/IP ACK packet from said client at said first BTCP module;

a5) receiving a HTTP request associated with said TCP/IP communication session; and

a6) storing said HTTP request and connection parameters associated with said TCP/IP SYN and TCP/IP ACK packets at said front-end node.

3. The method as described in Claim 2, wherein said step b) at said first BTCP module comprises the steps of:

b1) examining content of said HTTP request;

b2) determining which of said plurality of web server nodes, a selected back-end node, can best process said HTTP request based on said content;

b3) sending a handoff request message to a second BTCP module located at said selected back-end node over said control channel, if said selected back-end node is not said front-end node, said second BTCP module located below a second TCP module in a second operating system at said selected back-end node;

b4) including said connection parameters in said handoff request;

b5) including a first initial TCP state information for said first BTCP module, including said first ISN in said message; and

b6) receiving a handoff acknowledgment message from said second BTCP module if said TCP/IP communication session is successfully handed off.

4. The method as described in Claim 3, wherein said step c) at said second BTCP module comprises the further steps of:

c1) reconstructing said TCP/IP SYN packet using said connection parameters including changing a first destination IP address of said SYN packet to a second IP address of said selected back-end node;

c2) sending said TCP/IP SYN packet to said second TCP module;

c3) receiving a second TCP/IP SYN/ACK packet from said second TCP module;

c4) parsing a second initial TCP state from said second TCP/IP SYN/ACK packet, including a second ISN for said second TCP module, said second initial TCP state necessary for understanding said second TCP state for said second TCP module in said TCP/IP communication session;

c5) reconstructing said TCP/IP ACK packet using said connection parameters including changing a second destination IP address of said TCP/IP ACK packet to said second IP address;

c6) updating said TCP/IP ACK packet to reflect said second TCP state of said selected back-end node in said TCP/IP communication session;

c7) sending said TCP/IP ACK packet that is updated to said second TCP module; and

c8) sending a handoff acknowledgment message to said first BTCP module.

5. The method as described in Claim 4, wherein said step c) further comprises the steps of:

c9) migrating said first initial TCP state to said second BTCP module over said control channel by including said first initial TCP state in said handoff request message, said first initial TCP state including said first ISN, such that said second BTCP module can calculate said first TCP state for said front-end node in said TCP/IP communication session; and

c10) sending said second initial TCP state of said selected back-end node to said first BTCP module by including said second initial TCP state in said handoff acknowledgment message, said second initial TCP state including said second ISN, such that said first BTCP module can calculate said second TCP state for said second TCP module in said TCP/IP communication session.

6. The method as described in Claim 1, comprising the further steps of at said first BTCP module:

d) receiving incoming data packets from said client;

e) changing destination addresses of said incoming data packets to a second IP address of said selected back-end node;

f) updating TCP sequence numbers and TCP checksum in said data packets to reflect said second TCP state of said selected back-end node; and

f) forwarding said data packets to said selected back-end server computer.

7. The method as described in Claim 1, comprising the further steps of:

d) intercepting outgoing response packets from said selected back-end node at a second bottom TCP module located below a second TCP module in a second operating system at said selected back-end node;

e) changing source addresses of said response packets to a first IP address of said first front-end node;

f) updating sequence numbers and TCP checksum in said response packets to reflect said first TCP state; and

g) sending said response packets to said client.

8. The method as described in Claim 1, wherein said method is optimized for frequent TCP state handoffs.

9. The method as described in Claim 1, comprising the further steps of:

d) monitoring TCP/IP control traffic for said TCP/IP communication session at a second BTCP module located below a second TCP module in a second operating system at said selected back-end node;

e) understanding when said TCP/IP communication session is closed at said second server computer;

f) sending a termination message to said first server computer over said control channel;

g) terminating said TCP/IP communication session at said front-end node; and

h) freeing data resources associated with said TCP/IP communication session at said front-end node.

10. The method as described in Claim 1, wherein each node in said web cluster can perform as said front-end node and as said selected back-end node.

11. The method as described in Claim 1, if said selected back-end node is said front-end node, comprising the further steps of:

sending a reconstructed TCP/IP SYN packet from said first BTCP module to said first TCP module;

receiving a TCP/IP SYN/ACK packet at said first BTCP module from said first TCP module;

parsing a third initial TCP state from said second TCP/IP SYN/ACK packet, said third initial TCP state associated with a third TCP state for said first TCP module in said TCP/IP communication session;

updating a reconstructed TCP/IP ACK packet to reflect said third TCP state;

sending said updated TCP/IP ACK packet to said first TCP module;

updating incoming data packets from said client at said first BTCP module to reflect said third TCP state, including TCP sequences numbers and TCP checksum; and

updating outgoing response packets from said TCP module to reflect said first TCP state, including TCP sequence numbers and TCP checksum.

12. The method as described in Claim 1, wherein dynamically loadable modules, including said first BTCP

module, in operating systems at both said front-end node and said selected back-end node, including said first operating system, implement a TCP handoff protocol that works within kernel levels of an existing TCP/IP protocol.

13. In a communication network, a method of TCP state migration comprising the steps of:

a) establishing a TCP/IP communication session between a client computer and a first bottom TCP (BTCP) module located below a first TCP module in a first operating system at a front-end node, said front end node part of a plurality of web server nodes that form a web server cluster containing information, said TCP/IP communication session established for the transfer of data contained within said information;

b) receiving a HTTP request associated with said TCP/IP communication session at said first BTCP module;

c) examining content of said HTTP request;

d) determining which of said plurality of web server nodes, a selected back-end node, can best process said HTTP request based on said content;

e) handing off said TCP/IP communication session from said first BTCP module to a selected back-end node over a persistent control channel;

f) migrating a first TCP state of said first BTCP module to said selected back-end node, and sending a second TCP state of said selected back-end node to said first BTCP module over said control channel;

g) forwarding incoming data packets received at said first BTCP module to said selected back-end node; and

h) sending outgoing response packets from said selected back-end node directly to said client; and

i) terminating said TCP/IP communication session at said front-end node and said selected back-end node when said TCP/IP communication session is closed.

14. The method as described in Claim 13, wherein said step a) at said first BTCP module comprises the steps of:

a1) receiving a TCP/IP SYN packet from said client;

a2) selecting a first initial sequence number (ISN) for said first BTCP module that is associated with said TCP/IP communication session, said first ISN associated with a first TCP state of said first BTCP module;

a3) sending a TCP/IP SYN/ACK packet to said client;

a4) receiving a TCP/IP ACK packet from said client at said first BTCP module;

a5) receiving said HTTP request associated with said TCP/IP communication session from said client computer; and

a6) storing said HTTP request and connection parameters associated with said TCP/IP SYN and TCP/IP ACK packets at said front-end node.

15. The method as described in Claim 14, wherein said step e) at said first BTCP module comprises the steps of:

e1) sending a handoff request message to a second BTCP module located at said selected back-end node over

said control channel, if said selected back-end node is not said front-end node, said second BTCP module located below a second TCP module in a second operating system at said selected back-end node;

e2) including said connection parameters in said handoff request message;

e3) including a first initial TCP state information for said first BTCP module, including said first ISN in said handoff request message; and

e4) receiving a handoff acknowledgment message from said second BTCP module if said TCP/IP communication session is successfully handed off.

16. The method as described in Claim 15, wherein said step f) at said second BTCP module comprises the further steps of:

f1) reconstructing said TCP/IP SYN packet including changing a first destination IP address of said TCP/IP SYN packet to a second IP address of said selected back-end node;

f2) sending said TCP/IP SYN packet that is reconstructed to said second TCP module;

f3) receiving a second TCP/IP SYN/ACK packet from said second TCP module;

f4) parsing a second initial TCP state from said second TCP/IP SYN/ACK packet, including a second ISN for said second TCP module, said second initial TCP state necessary for understanding said second TCP state for said second TCP module in said TCP/IP communication session;

f5) reconstructing said TCP/IP ACK packet including changing a second destination IP address of said TCP/IP ACK packet to said second IP address;

f6) updating said TCP/IP ACK packet to reflect said second TCP state of said selected back-end node in said TCP/IP communication session;

f7) sending said TCP/IP ACK packet that is reconstructed and updated to said second TCP module; and

f8) sending a handoff acknowledgment message to said first BTCP module.

17. The method as described in Claim 16, wherein said step c) further comprises the steps of:

f9) migrating said first initial TCP state to said second BTCP module over said control channel by including said first initial TCP state in said handoff request message, said first initial TCP state including said first ISN, such that said second BTCP module can calculate said first TCP state for said front-end node in said TCP/IP communication session; and

f10) sending said second initial TCP state of said selected back-end node to said first BTCP module by including said second initial TCP state in said handoff acknowledgment message, said second initial TCP state including said second ISN, such that said first BTCP module can calculate said second TCP state for said second TCP module in said TCP/IP communication session.

17. The method as described in Claim 13, comprising the further steps of at said first BTCP module:

j) receiving incoming data packets from said client;

k) changing destination addresses of said incoming data packets to a second IP address of said selected back-end node;

l) updating TCP sequence numbers and TCP checksum in said data packets to reflect said second TCP state of said selected back-end node; and

m) forwarding said data packets to said selected back-end server computer.

18. The method as described in Claim 13, comprising the further steps of:

j) intercepting outgoing response packets from said selected back-end node at a second bottom TCP module located below a second TCP module in a second operating system at said selected back-end node;

k) changing source addresses of said response packets to a first IP address of said first front-end node;

l) updating sequence numbers and TCP checksum in said response packets to reflect said first TCP state; and

m) sending said response packets to said client.

19. The method as described in Claim 13, comprising the further steps of:

j) monitoring TCP/IP control traffic for said TCP/IP communication session at a second BTCP module located below

a second TCP module in a second operating system at said selected back-end node;

k) understanding when said TCP/IP communication session is closed at said second server computer;

l) sending a termination message to said first server computer over said control channel;

m) terminating said TCP/IP communication session at said front-end node; and

n) freeing data resources associated with said TCP/IP communication session at said front-end node.

20. The method as described in Claim 13, if said selected back-end node is said front-end node, comprising the further steps of:

sending a reconstructed TCP/IP SYN packet from said first BTCP module to said first TCP module;

receiving a second TCP/IP SYN/ACK packet at said first BTCP module from said first TCP module;

parsing a third initial TCP state from said second TCP/IP SYN/ACK packet, said third initial TCP state associated with a third TCP state for said first TCP module in said TCP/IP communication session;

updating said TCP/IP ACK packet to reflect said third TCP state;

sending said updated TCP/IP ACK packet to said first TCP module;

updating incoming data packets from said client at said first BTCP module to reflect said third TCP state, including TCP sequences numbers and TCP checksum; and

updating outgoing response packets from said TCP module to reflect said first TCP state, including TCP sequence numbers and TCP checksum.

21. The method as described in Claim 13, wherein each node in said web cluster can perform as said front-end node and as said selected back-end node, and said control channel allows for communication between all nodes for TCP state migration.

22. The method as described in Claim 13, wherein said plurality of server computers is coupled together over a local area network in said communication network.

23. The method as described in Claim 13, wherein said information is partitioned/partially replicated throughout each of said plurality of server computers.

24. A computer system comprising:

a bus;

a processor;

a memory coupled to said processor via said bus; and

a bottom TCP (BTCP) module comprising a dynamically loadable module that works within the kernel level of an existing TCP/IP protocol, wherein said memory contains instructions that when executed implement a method of TCP state migration between two nodes in a web server cluster, said web server cluster comprising a plurality of web server nodes, each of which comprising said computer

system, said method optimized for frequent TCP state handoffs, said method comprising steps of:

a) establishing a TCP/IP communication session between a client computer and a first BTCP module located below a first TCP module in a first operating system at a front-end node, said front end node part of said plurality of web server nodes that contain information, said TCP/IP communication session established for the transfer of data contained within said information;

b) handing off said TCP/IP communication session from said first BTCP module to a selected back-end node over a persistent control channel; and

c) migrating a first TCP state of said first BTCP module to said selected back-end node, and sending a second TCP state of said selected back-end node to said first BTCP module over said control channel.

25. The computer system as described in Claim 24, wherein said step a) of said method at said first BTCP module comprises the steps of:

a1) receiving a TCP/IP SYN packet from said client;

a2) selecting a first initial sequence number (ISN) for said first BTCP module that is associated with said TCP/IP communication session, said first ISN associated with a first TCP state of said first BTCP module;

a3) sending a TCP/IP SYN/ACK packet to said client;

a4) receiving a TCP/IP ACK packet from said client at said first BTCP module;

a5) receiving a HTTP request associated with said TCP/IP communication session; and

a6) storing said HTTP request and connection parameters associated with said TCP/IP SYN and TCP/IP ACK packets at said front-end node.

26. The computer system as described in Claim 25, wherein said step b) of said method at said first BTCP module comprises the steps of:

b1) examining content of said HTTP request;

b2) determining which of said plurality of web server nodes, a selected back-end node, can best process said HTTP request based on said content;

b3) sending a handoff request message to a second BTCP module located at said selected back-end node over said control channel, if said selected back-end node is not said front-end node, said second BTCP module located below a second TCP module in a second operating system at said selected back-end node;

b4) including said connection parameters in said handoff request;

b5) including a first initial TCP state information for said first BTCP module, including said first ISN in said message; and

b6) receiving a handoff acknowledgment message from said second BTCP module if said TCP/IP communication session is successfully handed off.

27. The computer system as described in Claim 26, wherein said step c) of said method at said second BTCP module comprises the further steps of:

c1) reconstructing said TCP/IP SYN packet using said connection parameters including changing a first destination IP address of said SYN packet to a second IP address of said selected back-end node;

c2) sending said TCP/IP SYN packet that is reconstructed to said second TCP module;

c3) receiving a second TCP/IP SYN/ACK packet from said second TCP module;

c4) parsing a second initial TCP state from said second TCP/IP SYN/ACK packet, including a second ISN for said second TCP module, said second initial TCP state necessary for understanding said second TCP state for said second TCP module in said TCP/IP communication session;

c5) reconstructing said TCP/IP ACK packet using said connection parameters including changing a second destination IP address of said TCP/IP ACK packet to said second IP address;

c6) updating said TCP/IP ACK packet to reflect said second TCP state of said selected back-end node in said TCP/IP communication session;

c7) sending said TCP/IP ACK packet that is reconstructed and updated to said second TCP module; and

c8) sending a handoff acknowledgment message to said first BTCP module.

28. The computer system as described in Claim 27, wherein said step c) of said method further comprises the steps of:

c9) migrating said first initial TCP state to said second BTCP module over said control channel by including said first initial TCP state in said handoff request message, said first initial TCP state including said first ISN, such that said second BTCP module can calculate said first TCP state for said front-end node in said TCP/IP communication session; and

c10) sending said second initial TCP state of said selected back-end node to said first BTCP module by including said second initial TCP state in said handoff acknowledgment message, said second initial TCP state including said second ISN, such that said first BTCP module can calculate said second TCP state for said second TCP module in said TCP/IP communication session.

29. The computer system as described in Claim 24, wherein said method at said first BTCP module comprises the further steps of:

d) receiving incoming data packets from said client;

e) changing destination addresses of said incoming data packets to a second IP address of said selected back-end node;

f) updating TCP sequence numbers and TCP checksum in said data packets to reflect said second TCP state of said selected back-end node; and

f) forwarding said data packets to said selected back-end server computer.

30. The computer system as described in Claim 24, wherein said method comprises the further steps of:

d) intercepting outgoing response packets from said selected back-end node at a second bottom TCP module located below a second TCP module in a second operating system at said selected back-end node;

e) changing source addresses of said response packets to a first IP address of said first front-end node;

f) updating sequence numbers and TCP checksum in said response packets to reflect said first TCP state; and

g) sending said response packets to said client.

31. The computer system as described in Claim 24, wherein said method comprises the further steps of:

d) monitoring TCP/IP control traffic for said TCP/IP communication session at a second BTCP module located below a second TCP module in a second operating system at said selected back-end node;

e) understanding when said TCP/IP communication session is closed at said second server computer;

f) sending a termination message to said first server computer over said control channel;

g) terminating said TCP/IP communication session at said front-end node; and

h) freeing data resources associated with said TCP/IP communication session at said front-end node.

32. The computer system as described in Claim 24, wherein in said method each node in said web cluster can perform as said front-end node and as said selected back-end node.

33. The computer system as described in Claim 24, if said selected back-end node is said front-end node, said method comprises the further steps of:

sending a reconstructed TCP/IP SYN packet from said first BTCP module to said first TCP module;

receiving a second TCP/IP SYN/ACK packet at said first BTCP module from said first TCP module;

parsing a third initial TCP state from said second TCP/IP SYN/ACK packet, said third initial TCP state associated with a third TCP state for said first TCP module in said TCP/IP communication session;

updating said TCP/IP ACK packet to reflect said third TCP state;

sending said updated TCP/IP ACK packet to said first TCP module;

updating incoming data packets from said client at said first BTCP module to reflect said third TCP state, including TCP sequences numbers and TCP checksum; and

updating outgoing response packets from said TCP module to reflect said first TCP state, including TCP sequence numbers and TCP checksum.